WHAT IS CLAIMED IS:

1. A microprocessor, comprising:

a branch unit to process branch instructions and provide a fetch unit with a next instruction address;

a load/store unit (LSU) to retrieve data from and stored data to a data memory of the microprocessor;

an arithmetic logic unit (ALU) to performing arithmetic operations on scalar, integer data; and

a vector unit to execute a vector instruction to perform a first operation on a first set of data operands and a second operation on a second set of operands, wherein the first and second operations differ.

2. The microprocessor of claim 1, wherein the vector unit comprises a primary register file and a secondary register, wherein an operand field in the vector instruction specifies a register within the primary register file and a corresponding register within the secondary register file.

3. The microprocessor of claim 2, wherein the vector instruction comprises fields for first, second, and third operand field, wherein the first, second, and third operand fields specify as many as three operands on which the first operation is to be performed and as many as three operands on which the second operation is to be performed.

4. The microprocessor of claim 3, wherein the vector unit includes a 3-input primary floating point unit and a 3-input secondary floating point unit, wherein the 3-input primary floating point unit is configured to perform a floating point operation on the first set of operands and the 3-input secondary floating point unit is configured to perform a floating point operation on the second set of operands.

5. The microprocessor of claim 4, wherein the 3-input primary floating point unit is configured to multiply first and third operands and further configured to add the second operand to or subtract the second operand from the resulting product.

6. The microprocessor of claim 4, where the vector unit is configured to permit either the primary register file or the secondary register file to provide the operands for the first, second, and third inputs to the primary floating point unit and the first, second, and third inputs to the secondary floating point unit.

7. The microprocessor of claim 2, wherein the vector unit is further characterized as being enabled to perform a cross instruction in which the first and second operations both use at least one operand from the primary register file and at least one operand from the secondary register file.

8. The microprocessor of claim 2, wherein the vector unit is further characterized as being enabled to perform a cross-replicate vector instruction in which the first and second operations are both performed using at least one common operand.

9. The microprocessor of claim 2, wherein the vector unit is configured to store a real portion of a complex number in the primary register file and an imaginary portion of the complex number in the secondary register file.

10. The microprocessor of claim 9, wherein the vector unit is configured to perform a complex operation in which the imaginary portion of a first operand is multiplied by an imaginary portion of a second operand in the first operation and in which the imaginary portion of the first operand is multiplied by a real portion of the second operand in the second operation.

11. A vector unit within a microprocessor, comprising:

means for receiving an instruction specifying first operand and second operands;

means for performing a first operation on a first set of data indicated by the first and second operands and a second operation on data indicated by the first and second operands, wherein the first and second operations differ.

12. The vector unit of claim 11, wherein the first operation is selected from an addition operation and an addition and negate operation and wherein the second operation is selected from a subtract operation and a subtract and negate operation.

13. The vector unit of claim 11, wherein the instruction is further characterized as specifying a third operand and wherein the first and second operations both include multiplying data specified by a pair of the operands.

14. The vector unit of claim 11, wherein each operand indicates a primary register of a primary register file and a corresponding secondary register of a secondary register file.

15. The vector unit of claim 11, wherein the primary register is configured to store a real portion of a complex number and the corresponding secondary register is configured to store an imaginary portion of the complex number.

16. A microprocessor including:

an execution unit enabled to execute an asymmetric instruction, wherein the asymmetric instruction includes a set of operand fields and an operation code (opcode);

wherein the execution unit is configured to interpret the opcode to perform a first operation on a first set of data indicated by the set of operand fields and to perform a second operation on a second set of data indicated by the set of operand fields, wherein the set of operand fields indicate different sets of data with respect to the first and second operations and further wherein the first and second operations are mathematically different.

17. The microprocessor of claim 16, further comprising a vector register file for use in executing the asymmetric instruction, wherein the register file includes a primary register file and a secondary register file, and wherein each operand in the set of operand fields indicates a register in the primary register and a corresponding register in the secondary register file.

18. The microprocessor of claim 16, wherein the instruction includes indicates first, second, and third operand fields indicating first and second sets of data on which the first and second operations are performed.

19. The microprocessor of claim 16, wherein each operand indicates a primary register in a primary register file and a corresponding secondary register in a secondary register file.

20. The microprocessor of claim 19, wherein the primary register file is configured to store a real portion of a complex number and wherein the secondary register file is configured to store an imaginary portion of a complex number.